باسمه تعالی


# تمرین (۲) درس برنامه نویسی پیشرفته

## ماشین حساب رشته ای

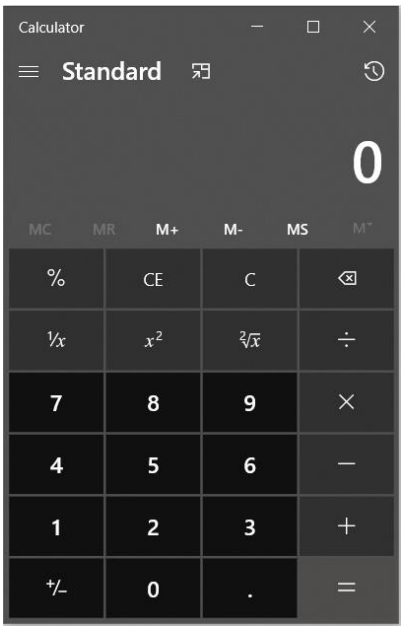
**استاد محترم**

**دکتر مرتضی یوسف صنعتی**

**بهار ۱۴۰۰**

## مقدمه

**قطعا همه شما با وسیله ای به نام ماشین حساب آشنا هستید. ماشین حساب وسیله ای برای انجام محاسبات ریاضی است که به طور کلی پس از تشخیص عملوندها و عملگر ها ، به محاسبه رشته های ورودی میپردازد.**




**عملگر و عملوند :**

**یک عملوند، هدف یک عملیات ریاضی است. هر عبارت که بین دو عملگر قرار بگیرد یا بعد از یک عملگر بیاید یک عملوند محسوب می شود.** (ویکی پدیا)

```
10 + 11
```

**در مثال فوق، علامت + یک عملگر است که برای عملیات جمع بکار برده شده است.**

## شرح جزئیات تمرین

در این تمرین یک ماشین حساب را پیاده سازی خواهید کرد اما این ماشین حساب ویژگی های بخصوص خود را دارد و با توجه به قوانین تمرین، نحوه انجام عملیات ها توسط این ماشین حساب متفاوت خواهد بود.

ماشین حساب رشته ای باید بتوند موارد زیر را محاسبه کند :

عملگر (+) :

این عملگر ابتدا دو عملوند را به یکدیگر متصل می کند و پس از مرتب کردن حروف، آن را به عنوان خروجی تحویل می دهد.

```
aebfc + gdh ⟶ abcdefgh
```

عملگر (-) :

این عملگر ابتدا طول هر دو عملوند اول و دوم را محاسبه و از یکدیگر کم می کند. خروجی این عملگر طولی از عملوند اول خواهد بود که اندازه آن از تفریق هر دو عملوند بدست آمده است.

```
abcde - xyz ⟶ 2 ⟹ abc
```

نکته * شماره خانه اول همه آرایه ها : ۰

نکته * عملوندی که از بقیه طول کمتری دارد باید از رشته های دیگر کم شود.

عملگر (*) :

این عملگر ابتدا دو عملوند را با یکدیگر مقایسه می کند و هر کدام که طول کمتری دارد را به تعداد طول دیگری تکرار می کند.

```
abcde * x ⟶ xxxxx
```

عملگر (/) :

این عملگر حروف بزرگ موجود در عملوند اول را با توجه به عملوند دوم،کوچک می کند.

```
ABCDEFG / adg ⟶ aBCdEFg
```

نکته * عملوند اول حاوی حروف بزرگ است

نکته * عملوند دوم باید حداقل ۱ کاراکتر باشد

**عملگر (<) :**

**این عملگر تمام حروف موجود در عملوند اول که بزرگ تر مساوی از عملوند دوم است ( از نظر ترتیب حروف الفبا ) را حذف می کند.**

```
abcdfabcdgabcdh < d ⟶ abcabcabc
```

**نکته * عملوند دوم فقط ۱ کاراکتر است**

**نکته * خروجی این بخش می تواند هیچ باشد**

**عملگر (&&) :**

**این عملگر بررسی می کند که آیا در هر دو عملوند حروف صدا دار وجود دارد ( True ) یا نه ( False )**

```
a && a ⟶ True
```

```
bcd && xyz ⟶ False
```

```
a && b ⟶ False
```

**عملگر (||) :**

**این عملگر بررسی می کند که آیا در هر دو عملوند حداقل یک حرف صدا دار وجود دارد ( True ) یا نه ( False )**

```
a || a ⟶ True
```

```
a || b ⟶ True
```

```
bcd || xyz ⟶ False
```

**نکته * حروف صدا دار انگلیسی :** a e i o u

**عملگر (=!) :**

**این عملگر،عملوند اول را حول یک محور یا همان عملوند دوم می چرخاند.**

```
abcdefgh =! e ⟶ fgheabcd
```

**نکته * عملوند دوم فقط ۱ کاراکتر است**

**عملگر (=+) :**

**این عملگر عملوند دوم را به تک تک حروف عملوند اول اضافه می کند.**

```
abcd += x ⟶ axbxcxdx
```

**نکته * عملوند دوم فقط ۱ کاراکتر است**

**عملگر (=-) :**

**این عملگر،عملوند دوم را در هر قسمت از عملوند اول بیابد،حذف می کند .**

```
axbxcxd -= x ⟶ abcd
```

```
abcd -= x ⟶ abcd
```

```
abcd -= abcd ⟶ null
```

**نکته * عملوند دوم باید کوچکتر یا مساوی عملوند اول باشد**

**تاریخچه :**

**این ماشین حساب دارای تاریخچه است به طوری می تواند خروجی حاصل از تمامی عملیات های انجام شده در گذشته را به صورت لیست نمایش دهد.**

```
> history

> abcdefgh
> abc
> xxxx
> aBCdEFg
> abcabcabc
> True
> False
> False
> True
> True
> False
> fgheabcd
> null
```

**نکته * برای نگهداری تاریخچه از وکتور استفاده کنید**

**نکته * دستوری که تاریخچه را نشان می دهد :** > history

## قوانین پیاده سازی تمرین

**توضیحات و نکات کافی برای پیاده سازی تمرین در این بخش وجود دارد :**

**ورودی های این برنامه در قالب زیر است که باید در ابتدا در یک آرایه ریخته شود و با متد مناسبی از یکدیگر جدا شوند.**

```
array ⟵ Operand Operator Operand
```

```
[Operand] [Operator] [Operand]
```

**در پیاده سازی این تمرین تمام ورودی ها نهایتا ۱۰ کار اکتری هستند. (دقت کنید برای فهم بهتر کارکرد عملگرها،ورودی های بیشتر از ۱۰ کاراکتر هم مثال زده شد و هدف ، صرفا بیان موضوع بود و به این معنی نمی باشد که برنامه ورودی های بیشتر از ۱۰ کاراکتری را پشتیبانی می کند) و باید با توجه به اینکه هر ورودی چه اندازه ای دارد، شئ مناسب آن ساخته شود.**

**در پیاده سازی باید ده کلاس وجود داشته باشد. (که یکی از کلاس ها فقط رشته های تک کاراکتری مورد نیاز برای ماشین حساب را ایجاد می کند. یکی دیگر از کلاس ها ، رشته های دو کاراکتری و به همین ترتیب...)**

**دقت کنید عملوندهای عملگرهای مورد نیاز برای این ماشین حساب فقط شئ هایی هستند که از کلاس های موجود در برنامه (که بالاتر ذکر شد) ساخته می شوند و عملگرهای این ماشین حساب با هر نوع رشته دیگری کار نمی کند و رشته های ورودی کلاس ها باید داده عضو باشند.**

**بنابراین بعد از اینکه ورودی از کاربر دریافت شد، که در آرایه مدیریت شد،و سایز عملوند ها تشخیص داده شد، متناسب با سایز عملوندها شئ های رشته ای مورد نیاز متناظر با آنها از کلاس های موجود ساخته شوند.**

**(تعریف مناسب کلاس ها، از جمله تشخیص اجزای آن و نوع کانستراکتور و توابع مورد نیاز دیگر کلاس ها به عهده دانشجویان می باشد.)**

**در این تمرین باید برای عملگر های تعریف شده ، توابع مناسبی پیاده سازی شود که تابع متناظر با هر عملگر بسته به نوع عملیاتش شئ های مربوط به عملوند های اول و دوم را دریافت میکند. این توابع ، توابع عضو نیستند و خارج از کلاس ها باید تعریف شوند .**

**دقت کنید برای هر عملگر به یک تابع نیاز داریم بنابراین تشخیص اینکه توابع چگونه تعریف شوند که پارامتر های آن بتوانند هر نوع شئ ایجاد شده از کلاس های موجود در برنامه را پوشش دهند، و خروجی توابع بر عهده دانشجویان می باشد.**

## بخش امتیازی

**در این بخش با پیاده سازی موارد بیشتر، مورد تشویق قرار خواهید گرفت.**

**سوال : آیا می توانید چهار عمل اصلی تعریف شده در بالا را برای بیشتر از دو عملوند پیاده سازی کنید ؟**

**- مثال از نحوه پیاده سازی :**

```
ae + cfb + xy + dghz ⟶ abcdefgxyz
```

```
abascd - abcadkj - abc ⟶ abas abcad
```

```
abcds * xyz * bb * x ⟶ xxxxx xxx xx
```

```
ABCD / AHAHAHA / NASA / na ⟶ aBCD aHaHaHa naSa
```

### نکات کلی

**- توجه کنید موارد بالا تنها چند مثال برای درک بهتر قابلیت های ماشین حساب است و ماشین حساب ما باید بتواند تمامی حالت های مختلف را ( در چهارچوب گفته شده ) پشتیبانی کند.**

**- تمامی عملیات های ماشین حساب دارای دو عملوند می باشد نه بیشتر !**

**- در تمام پیاده سازی ها باید به استثناء ها ( exception ) رسیدگی کرده و آن ها را بررسی کنید.**

**- برای نمایش نتایجی که خالی هستند می توانید از کلمه null در ترمینال استفاده کنید.**

**- رشته ها می توانند حاوی حروف انگلیسی یا اعداد باشند .**

**- استفاده از cmake اجباری است .**

**- نوشتن گزارش کار به صورت کامل و مرتب اجباری است.**

**- کامنت گذاری و ماژولاریتی تمرین رعایت شود .**

**- دریافت تمرین ممکن است به صورت حضوری و آنلاین انجام شود.**

**مهلت ارسال تمرین**

**نهم فروردین ماه ۱۴۰۰ لغایت پانزده فروردین ماه ۱۴۰۰**